A Simple Mechanism to Adapt Leakage-Control Policies to Temperature

Stefanos Kaxiras, Polychronis Xekalakis, Georgios Keramidas
{kaxiras, xekalakis, keramidas}@ee.upatras.gr

Department of Electrical and Computer Engineering,
University of Patras, Greece

ABSTRACT

Leakage power reduction in cache memories continues to be a critical area of research because of the promise of a significant pay-off. Various techniques have been developed so far that can be broadly categorized into state-preserving (e.g., Drowsy Caches) and non-state preserving (e.g., Cache Decay). Decay saves more leakage but also incurs dynamic power overhead in the form of induced misses. Previous work has shown that depending on the leakage vs. dynamic power trade-off, one or the other technique can be better. Several factors such as cache architecture, technology parameters and temperature, affect this trade-off. Our work proposes the first mechanism —to the best of our knowledge— that takes into account temperature in adjusting the leakage control policy at run time. At very low temperatures, leakage is relatively weak so the need to tightly control it is not as important as the need to minimize extra dynamic power (e.g., decay-induced misses) or performance loss. We use a hybrid decay+drowsy policy where the main benefit comes from decaying cache lines while the drowsy mode is used to save leakage in long decay intervals. To adapt the decay mode to temperature, we propose a simple triggering mechanism that is based on the principles of decaying 4T thermal sensors and, as such, tied to temperature. The hotter the cache is, the faster cache lines are decayed since it is beneficial to do so with very high leakage currents.Conversely, when the cache temperature is low, our mechanism defers putting cache lines in decay mode to avoid dynamic power overhead but still saves a significant amount of leakage using the drowsy mode. Our study shows that across a wide range of temperatures, the simple adaptability of our proposal yields consistently better results than either the decay mode, or drowsy mode alone, improving over the best by as much as 33%.

Categories and Subject Descriptors

C.1 [Other Architecture Styles]: Adaptable Architectures

ISLPED'05, August 8–10, 2005, San Diego, California, USA.
Copyright 2005 ACM 1-59593-137-6/05/0008...$5.00.

General Terms

Performance, Design

Keywords

Cache Decay, Drowsy Cache, Thermal Adaptation,Hybrid Leakage Mechanism.

1. INTRODUCTION

Leakage power has been identified as a critical problem for deep sub-micron technologies. Since leakage power is consumed by every transistor regardless of switching, it is desirable to confront it in as many transistors as possible. Caches, of course, comprise the bulk of the transistor budget in modern processors and although memory cell leakage is not as pronounced as logic gate leakage [20], their overall contribution makes them a prime target for power optimizations.

Several techniques have been developed so far to combat leakage in caches. One of the first techniques proposed is the gated-Vdd technique [3] (and subsequently gated-Vss) which disconnects a cell from its supply voltage (ground). Subthreshold leakage falls to very low levels, but the cell's contents are lost. Various policies have been proposed to apply gating to cells such as the DRI cache [3] (where gating is applied to large parts of the cache, causing it to resize), and Cache Decay [4] (where gating is applied to individual cache blocks deemed no longer useful). Regardless of the policy used, gated techniques are non-state preserving. This invariably leads to more accesses to lower memory hierarchy levels, which in turn translate to increased dynamic power consumption. Thus, there is a limit to how aggressively one can use Vdd/Vss gating before leakage benefits are offset by dynamic power increase and performance loss.

State preserving techniques such as drowsy caches, put cells into low Vdd mode without destroying their contents. In this mode the cells leak significantly less but, still, more than the gated Vdd/Vss approach. Drowsy techniques do not entail extraneous misses but can incur performance loss. Accessing drowsy cells requires additional latency —to bring them back to full Vdd— but typically less than the latency of a decay-induced miss.

Thus, there is complementarity between the two techniques. The potential gain of using decay is higher but the penalty of discarding the wrong line prematurely is considerable. On the other hand, we do not gain as much from drowsy lines, but we can be more aggressive in putting lines to drowsy mode since there is no immediate dynamic power penalty. A hybrid scheme that first puts a line in drowsy mode and subsequently, after some period of inactivity, in decay mode seems to capture the best of the two techniques. This is also proposed in recent work that examined such schemes using oracle knowledge [6].

The timing for entering the decay mode in a hybrid scheme is dictated by the leakage to dynamic power ratio. Leakage currents increase exponentially with temperature, so in high temperatures it is beneficial to put lines in decay mode earlier, even with increased decay-induced misses. In contrast, at low temperatures, leakage can be dealt more effectively with the drowsy mode, deferring passage to the decay mode. Under low temperature conditions, decay-induced misses are far more damaging since the relative benefit from leakage control is smaller.

One of the main contributions of this paper is the design of a timing mechanism that automatically takes into account temperature-induced variations in leakage currents. Although previous work has studied leakage at various temperatures, this is the first proposal that adjusts decay intervals with temperature. Our design is based on decaying 4T cells which act as timers—the principle behind decaying 4T thermal sensors [5]. When they decay they force the corresponding cache line into decay mode. Their timing is designed for high temperature conditions where they provide maximum *normalized* leakage power reduction under desired performance constraints. At low temperature conditions, where the benefit from leakage reduction is smaller, our mechanism puts emphasis on lowering the dynamic overhead and automatically stretches the time to go into decay mode.

To summarize, the contributions of our work are:

- We are studying an adaptive hybrid drowsy+decay design that adapts the decay mode according to temperature and uses the drowsy mode to save leakage in long decay intervals. Our intent is to show that even a simple adaptive scheme works well and surpasses all non-adaptive schemes.
- The simplicity of our proposal stems from an inexpensive timing mechanism based on 4T decaying cells. This mechanism is less complex than a digital scheme based on hierarchical counters as in [4] and naturally adapts to temperature.

Structure of this paper—In the rest of this paper we present related work in Section 2, our adaptive scheme in Section 3, followed by details on the experimental methodology in Section 4 and the simulation results in Section 5. We conclude with a summary in Section 6.

2. RELATED WORK

Cache Decay, utilizes hierarchical counters to detect possibly unneeded cache lines. A global counter provides a signal for local per-cache-line counters. A cache line is gated if its local counter saturates without any intervening access to the line. Kaxiras et al [4] has shown that an average of 67% of static power consumption can be saved with a minimal performance loss (due to decay induced misses). Furthermore, several papers [4,13,14] have shown that by adapting the decay interval to individual applications one can set an upper bound to the performance loss, minimizing dynamic power overhead. As leakage increases relatively to dynamic power, gating of the appropriate cache lines yields increasingly better results.

In the state preserving camp, one of the first leakage reduction mechanisms proposed is the Drowsy Cache [2]. Since the cost of waking up a drowsy block is small (about 7 cycles), a sensible and easy way to implement this approach is to put all of the cache in the drowsy state periodically. Flautner et al [2] show that a Drowsy Cache using this simple policy (with a 1Kcycle period) achieves 54% leakage power reduction with a performance loss of no more than 1.2%.

At the circuit level, Cache Decay uses the stack effect [9] while the Drowsy Cache employs a Voltage Scaling (DVS) approach [10]. To reduce the leakage power, other circuit techniques can be used such as MTCMOS, DTCMOS, Reverse Body Bias and larger than Vdd Forward Body Bias [15,16,17,18]. Borkar et al [11] have evaluated some of these techniques and have shown that the stack effect is the most effective means to reduce leakage power, but because it lowers the active current in the normal mode operation, it is also the slowest. They have also shown that lowering voltage is inferior in terms of both leakage savings as well as in speed (low voltage underrates Ion significantly). Thus, they conclude that RBB is the best compromise between leakage savings and speed in normal operation mode. However, the usage of RBB requires the generation and routing of extra power supply to the body and well terminals of n- and p-MOS transistors. In addition, it requires the usage of a triple-well bulk CMOS process [11] increasing the overall implementation cost. MTCMOS is in effect a dynamic implementation of the RBB scheme, which makes it more costly in terms of fabrication. Its advantage is the capability to dynamically switch threshold voltages, but unfortunately it lacks the switching speed of the DVS and gated-Vdd approaches. DCMOS is an extreme case of the MTCMOS but the fact that it is a static approach makes it impractical in terms of speed. Thus, the best compromise in ease of fabrication, switching speed between high and low leak modes, and leakage savings is DVS —for state-preserving techniques— and gated Vdd (Vss) —for the non-state preserving techniques.

Li et al examined the different energy savings for L1 data caches for the Drowsy and Cache Decay mechanisms [7]. Their work debunks a common belief that state preserving techniques are superior to non-state preserving ones. More specifically, for a fast L2 cache (5-8 cycles latency), Cache Decay is better in terms of both performance loss and energy savings than the Drowsy cache. However, in our work, we use a slower L2 that does not particularly benefit Decay.

Recently, Meng et al [6] have shown that under oracle knowledge of the access stream, the best approach is a hybrid gated + drowsy scheme. Although their theoretical results are indicative of how well a hybrid scheme could work, their work was more targeted at discovering the limits to which these techniques are applicable. Furthermore the dependence of static power to temperature is not accounted for —the trade-off between drowsy and decay modes cannot be static with temperature.

3. HYBRID, TEMPERATURE-AWARE MECHANISMS

The leakage control mechanism in our work is a hybrid drowsy + decay scheme. The decay mode is used for maximum leakage power savings. To avoid mistakes, we decay a cache line only after a certain *long* period —*a decay interval*— of inactivity.

The decision on how long to wait to enter the decay mode depends on the relative strength of the leakage power to dynamic power. Technology characteristics such as technology node, supply voltage, threshold voltage, and gate oxide thickness, significantly affect the strength of leakage currents. As we move to smaller feature sizes (e.g, 130, 90, 70 nm) and to lower supply voltages, leakage increases exponentially. On the other hand, the dynamic power component is affected by the number of induced misses which is largely determined by the cache architecture.

Most importantly, for a given cache (fixed design and technology parameters) leakage varies substantially with temperature at *run-time*. At high temperatures (high leakage), we want to be aggressive in using the decay mode since, even with an increase in dynamic power overhead, we maximize overall power savings. But when leakage currents are low, the dynamic power overhead can dominate, regardless of the percentage of leakage we save.

With decay mode alone, waiting in active mode for the duration of the decay interval is a necessary cost to pay. However, in a hybrid scheme, the decay interval is exploited by the drowsy mode. The trade-off in drowsy mode is markedly different than in decay mode: we can enter and leave the drowsy mode much more cheaply and the penalty of getting it wrong is much less: there is no immediate dynamic power overhead but only an increase in latency. Thus, we use the drowsy mode within the confines of the decay interval to reduce leakage power significantly over the decay mode alone. To minimize performance overhead associated with the drowsy mode we wait for a period of inactivity (*drowsy interval*) before putting a cache line in the drowsy mode. Meng et al [6] showed that with oracle knowledge (no performance penalty) we only have to wait a few cycles before putting a line in drowsy mode but in reality we have to wait on the order of 1Kcycles to avoid excessive performance loss [2].

Our adaptive timing mechanism stretches the decay interval at lower temperatures. This leads to significantly less leakage reduction from the decay mode but also minimizes its dynamic power overhead. Since the drowsy mode is engaged more at low temperatures (within the longer decay intervals) the net effect is an overall reduction of the total power across a wide range of temperatures. To keep our scheme simple and to bound performance loss we do *not* adapt the drowsy interval which is fixed for all temperatures. More sophisticated techniques could adapt both the decay and the drowsy interval simultaneously but for diminishing additional gains.

There are several methods to increase the decay interval with lower temperatures. With a hierarchical counter mechanism such as the one in [4], a thermal sensor controls the global counter that advances the local counters. Increasing the magnitude of the global counter increases the decay intervals proportionally. This would give us fine control over the decay intervals but it is also costly to implement. In this paper we go one step further and propose a very efficient timer per cache line: a decaying quasi-static memory cell.

3.1 Decaying 4T Timers

We have previously proposed the 4T DRAM cell as a temperature-sensitive timer in [5] where the frequency at which it ticks is a measure of temperature. Our mechanism is based on the same idea: the decay interval of our cache lines is regulated by the decay of a 4T cell.

Figure 1 depicts the architecture of the complete timing mechanism that sets a cache line in drowsy or decay mode. Implemented adjacent to each cache line, this mechanism adapts to the temperature of the line’s immediate surrounding area. To keep the drowsy interval fixed, we use a single-bit local counter implemented as a small FSM. For the drowsy mode we have ascertained that we do not need higher resolution in the local counter. The counter is reset with every access and advances with an external signal. If it reaches its saturating state without an intervening access, it puts the line in drowsy mode using the voltage regulator depicted in Figure 2.

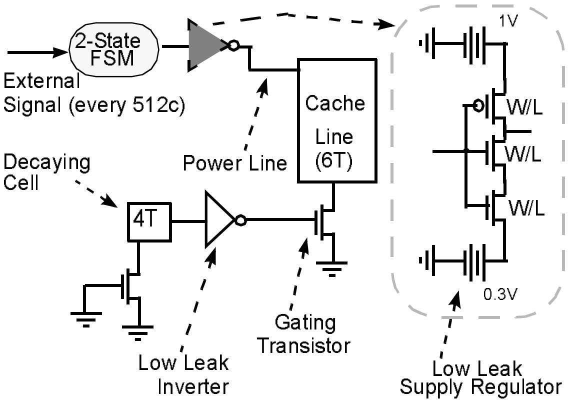

Figure 1. Decaying 4T timer for a hybrid decay+drowsy policy.

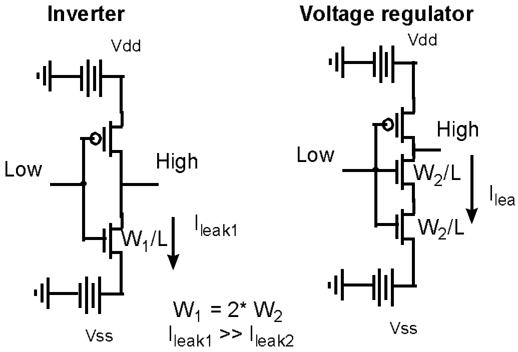

Figure 2. Low Leak Inverters - Voltage Regulators

Accessing the line, also charges (writes a “1”) the 4T cell. As long as the 4T holds “1” the cache line is connected to ground. If, however, the 4T is left un-accessed for a long period, it decays and gates the cache line via a low leak inverter (shown in Figure 2). As soon as the line is accessed again the 4T reinstates the connection to ground.

3.2 4T cell Temperature Characteristic

A 4T cell’s *retention* or *decay* time is a function of temperature. Retention times increase exponentially with lower temperatures, something expected from the relationship between temperature and leakage.

Since the retention times of a simple 4T cell are very small for our purposes [5], we gate it with another transistor. Due to the stack effect, the leakage currents of the cell are reduced, which increases retention times. If, however, we need even larger retention times we employ the non-minimum gate length approach [12] to the design of the cell itself to adjust its leakage. The size of the gating transistor is not a concern since

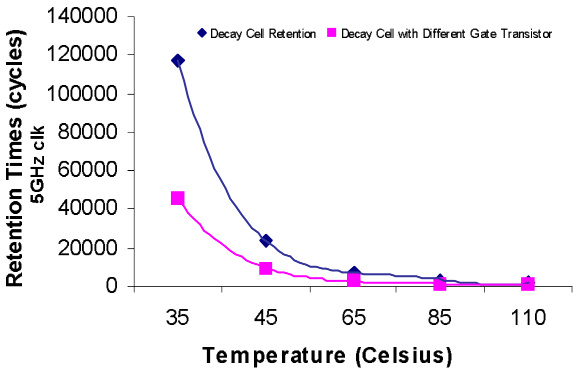

Figure 3. Retention Times vs. Temperature for two cells with different gating transistors

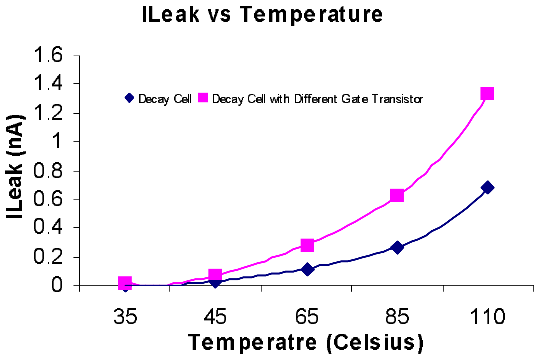

Figure 4. Leakage Currents vs. Temperature for two 4T Cells with different gating transistors

it can be shared among many 4T cells. Retention times of two cells with different gating transistors (in terms of gate length) are shown in Figure 3. The difference in retention times stems from the difference in the leakage currents shown in Figure 4.

A notable effect of the 4T cell —also observed in [5]— is that, as temperature rises, retention times converge to the same value. This means that decay intervals do not decrease uncontrollably with increasing temperatures, something that could be catastrophic, leading to excessive dynamic power overhead and performance loss. As we show in our results (Section 5) 4T decay exhibits exactly the desired adaptability we seek. Finally, retention times of 4T cells are remarkably insensitive to process variations (e.g., L_{drawn}, T_{ox}) and to noise [5] making it easier to design cells with the desired retention times.

4. METHODOLOGY

4.1 Processor Model and Benchmarks

To evaluate the effectiveness of our proposal, from an architectural point of view, we performed simulations using the HotLeakage simulator [19]. Hotleakage is a detailed cycle level simulator which dynamically tracks static and dynamic power for each CPU structure (e.g., caches).

The processor model we use is modeled after the Alpha 21264 [8]. The execution core is a 4-wide superscalar pipeline and the memory hierarchy includes a 64KB, 2-way set-associative L1 instruction cache with a single cycle hit latency, a 64 KB, 2-way set-associative L1 data cache with a 3 cycle hit latency, a unified 2 MB L2 cache, with 4-way set-associativity. This configuration reflects prior work that examined the trade-off between decay and drowsy modes at various temperatures [7]. We chose to use an 11 cycle latency for the L2 that according to [7] benefits the drowsy mode over decay. Finally, we use process parameters for a 70 nm technology with a Vdd of 0.9V and a 5GHz clock. Our processor and technology parameters also reflect the ones used in [7].

The benchmark suite for this study consists of a set of six SPEC2000 benchmarks, applu, gzip, gcc, mesa, vortex, and vpr, compiled for the Alpha AXP ISA. For each program, we skip the first billion committed instructions to avoid unrepresentative startup behavior at the beginning of the program’s execution, and then simulate 200 million committed instructions using the reference input set. We chose these benchmarks for two reasons: they are frequently used in the computer architecture literature and they are singled out in other leakage power reduction papers [6][7].

To design the 4T decaying timers we use accurate SPICE simulations with the BSIM4 predictive transistor models [1].

5. RESULTS

In this section we present a comparative study of the decay scheme, the drowsy scheme, and our proposed adaptive-hybrid scheme for various temperatures. Because of the volume of information we do not show results for individual benchmarks but rather present the averages over all benchmarks. In general, the behavior of individual benchmarks does not deviate from the averages and there are no inversions of the overall trends.

Figure 5 shows how Cache Decay varies with temperature (worse at 35^{o}, best at 110^{o} C, because of the changing leakage-to-dynamic-power ratio) and for decay intervals ranging from 800 cycles to 102400 cycles. The vertical axis shows *normalized leakage*, i.e., the sum of the remaining leakage at that temperature plus the dynamic power overhead, divided by the original leakage at the same temperature:

$$NormalizedLeakage = \frac{NewLeakage + DynamicPowerOverhead}{OriginalLeakage}$$

The decay interval trade-off is obvious in the graph. For 110^{o} C, the decay interval that minimizes normalized leakage is about 1600 cycles, while for 35^{o} increases to about 6400 cycles. The performance impact of the varying decay intervals is shown in Figure 6 (note that performance impact does not depend on temperature). Clearly, small decay intervals are prohibitive if we consider energy-delay products since performance loss can wipe-out the benefit from leakage savings. Thus, to limit performance loss to 2% we do not consider decay intervals smaller than 3200 cycles for our schemes.

On the same graph (Figure 5) we plot normalized leakage for a pure drowsy scheme with a fixed drowsy interval of 512 cycles —the choice of the drowsy interval is discussed below. Its normalized leakage appears flat (since it has no relation to the *decay* interval) but varies —imperceptibly in Figure 5— with temperature. This variation is due to the leakage overhead of the timing mechanisms which scales disproportionately with temperature.

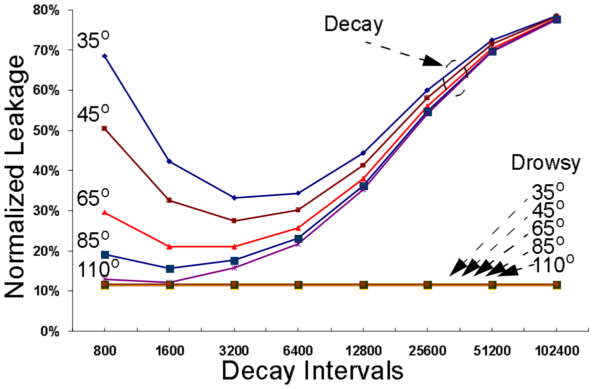

Figure 5. Decay vs. drowsy: Decay varies with temperature but is always worse than drowsy for the specific configuration.

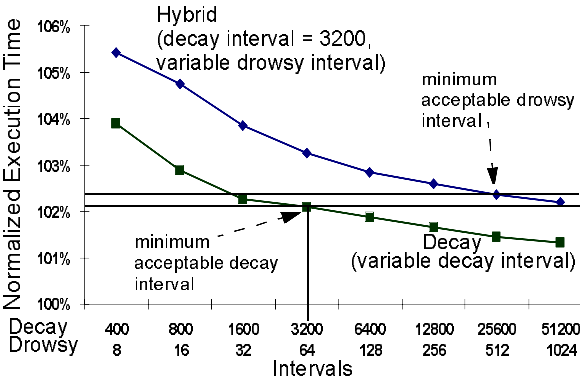

Figure 6. Performance impact of decay and hybrid schemes with varying decay/drowsy intervals

To determine the fixed drowsy interval for our schemes —the waiting time before inactive lines are put in drowsy mode— we examine its performance impact on a hybrid scheme which already uses the minimum (acceptable) decay interval of 3200 cycles. We chose the minimum drowsy interval that limits performance loss (execution time increase) of the hybrid scheme to 0.5% over the corresponding decay scheme. Figure 6 shows that a drowsy interval smaller than 512 cycles incurs performance loss for the hybrid scheme greater than our threshold. Hence, we chose 512 cycles for the drowsy interval. This makes the drowsy scheme more effective than in other work where the minimum drowsy interval typically is 1Kcycles [7][2].

Overall, we chose parameters that favor the drowsy mode at all temperatures (Figure 5). With a different set of parameters, decay can do better [7] but in such cases, a hybrid scheme would be better than both by default. In contrast, we show that a hybrid scheme outperforms both drowsy and decay schemes even when the decay scheme is significantly worse than the drowsy scheme (e.g., at 35^{o} in Figure 5).

Figure 7 shows normalized leakage for the hybrid scheme varying the decay interval. In the same graph we also plot the normalized leakage of the drowsy scheme. Choosing the right decay intervals at the right temperatures the hybrid scheme easily outperforms the drowsy scheme. As we argued, the

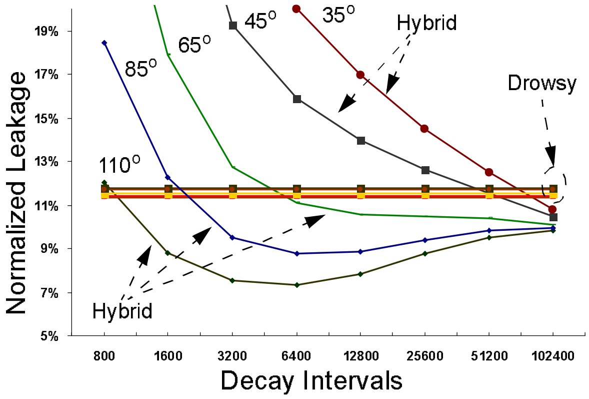

Figure 7. Hybrid (decay+drowsy) vs. drowsy: choosing the right decay interval, hybrid outperforms drowsy at all temperatures.

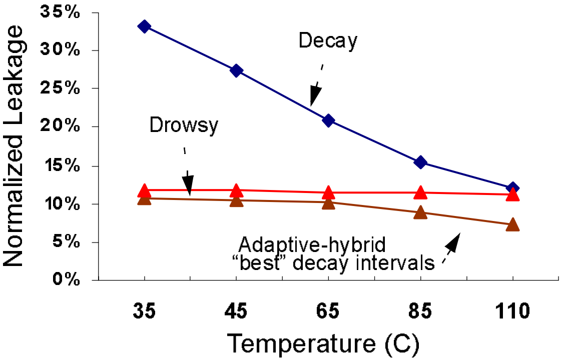

Figure 8. Hybrid with "best" decay intervals vs. decay vs. drowsy: Hybrid consistently outperforms both for all temperatures

decay interval must be short at high temperatures and much longer at low temperatures.

Although determining optimal decay intervals for every temperature is beyond the scope of this paper, the decay intervals we examined give us a good idea of the range of values where the optimal interval can be found (Table 1). Since we don't have the actual optimal intervals, we use the best —lowest points of the hybrid curves in Figure 7— of the decay intervals as an approximation (Table 1). Pseudo-adapting our hybrid scheme using these "best" decay intervals we obtain the results shown in Figure 8. This figure contrasts the hybrid scheme to the drowsy scheme and to the best cases —lowest points of the decay curves in Figure 5— for the decay scheme. The hybrid scheme consistently yields lower normalized leakage than both other schemes, improving over the drowsy scheme by 10% in 35^{o} C and by 33% in 110^{o} C.

With our 4T temperature-adaptive timers we aim to approximate the sequence of optimal decay intervals. Table 1 shows the range of values that contains the optimal decay interval at various temperatures. We designed our 4T timer to give 3600 cycles decay time at 110^{o} C. The resulting decay times of the 4T for the other temperatures are shown in the right most column of Table 1. Using, now, the 4T decay intervals in the hybrid scheme we obtain the results shown in Figure 9. Our 4T adaptive hybrid scheme tracks well the

Table 1. Ranges for optimal decay intervals, best decay intervals we examined, and 4T decay intervals at various temperatures

Temperature	Range for Optimal Decay IntervaL	"Best" Decay Interval examined	4T Decay Intervals
35	> 102400	102400	261430
45	> 102400	102400	53495
65	> 51200	102400	13025
85	6400-12800	6400	6355
110	3200-6400	3200	3600

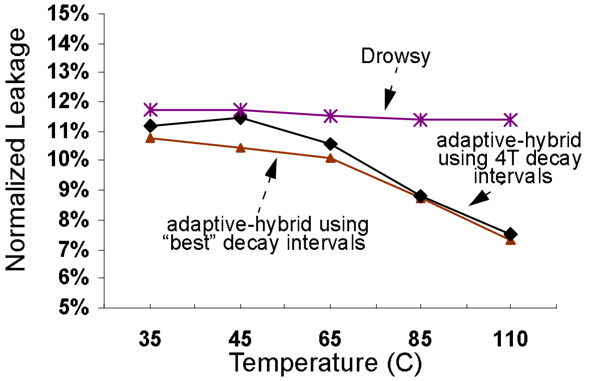

Figure 9. Hybrid with "best" decay intervals vs. Hybrid with 4T decay intervals vs. drowsy: Hybrid with 4T tracks well Hybrid with "best" and consistently outperforms drowsy

results with the best decay intervals from Figure 7, especially at high temperatures. At 110^{o} C our hybrid scheme saves 92% of the leakage power by using the decay mode aggressively.

6. CONCLUSIONS

We present the first —to the best of our knowledge— temperature-aware leakage control mechanism. We combine the decay and drowsy schemes into a hybrid scheme in which the decay mode is adapted to temperature. At high-temperature, high-leakage conditions our scheme employs the decay mode aggressively to maximize leakage savings while at low-temperature, low-leakage conditions it employs the drowsy mode much more to minimize dynamic power overhead from the decay mode. We achieve this by keeping the drowsy interval fixed and simply adjusting the decay interval to temperature. For this purpose we propose using decaying 4T cells as decay timers. They exhibit exactly the adaptive behavior we seek and their decay intervals can be designed to approximate a sequence of optimal decay intervals. Using 4T timers, our hybrid scheme consistently outperforms the best of the non-adaptive schemes for all temperatures and by as much as 33%.

7. ACKNOWLEDGEMENTS

We would like to thank Yan Meng and Tim Sherwood for their helpful feedback. This work is supported by the HiPEAC Network of Excellence IST-004408 and by Intel Research Equipment Grant #15842.

8. REFERENCES

[1] Weidong Liu, et al. "BSIM3v3.2 MOSFET Model User's Manual," Dept. of EE and CS, U.C. Berkeley.

[2] K. Flautner, et al. "Drowsy Caches: Simple Techniques for Reducing Leakage Power," In *Proc. ISCA 29, 2002.*

[3] Michael Powell et al. "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories," In *Proc. ISLPED 2000.*

[4] S. Kaxiras, Z. Hu, M. Martonosi, "Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power," In *Proc. ISCA 28*, 2001.

[5] S. Kaxiras, Polychronis Xekalakis "Decaying 4T Thermal/Leakage Sensors," In Proc. ISLPED, 2004.

[6] Yan Meng, et al. "On the Limits of Leakage Power Reduction in Caches," In Proc. HPCA-11, 2004.

[7] Yingmin Li et al. "State-Preserving vs. Non-State-Preserving Leakage Control in Caches," In Proc. of the 2004 DATE Conference, pp. 22-27, Feb. 2004.

[8] R. E. Kessler, et al. "The Alpha 21264 microprocessor architecture," In *Proc. ICCD 1998*, pages 90–95, Oct. 1998.

[9] Y. Ye, **S.** Borkar, and V. De, "A Technique for StandbyLeakage Reduction in High-Performance Circuits," *Symp.ofVLSI Circuits*, pp. 40-41, 1998.

[10] T. Pering, et al. "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms," In Proc. ISLPED 1998.

[11] Bhaskar Chatterjee, et al. "Effectiveness and Scalling Trends of Leakage Control Techniques for Sub-130nm CMOS Technologies," In Proc. ISLPED 2003.

[12] N. Sirisantana, L. Wei, and K. Roy, "High-Perfomance Low-Power CMOS Circuits Using Multiple Channel Length and Multiple Oxide Thickness," in Proc. ICCD, 2000.

[13] H. Zhou, et al. "Adaptive mode control: A static-power-efficient cache design," In Proc. PACT 2001, Sept. 2001.

[14] S. Velusamy, et al. "Adaptive cache decay using formal feedback control," In Proc. WMPI-2, May 2002.

[15] Hari Ananthan, et al. "LargerthanVdd Forward Body Bias in Sub0.5V Nanoscale CMOS," In Proc. ISLPED, 2004.

[16] S. Mutah, et al. "1-V Power Supply High-Speed Digital Circuit Technology with Multi-Threshold Voltage CMOS," IEEE Journal of Solid-State Circuits, 30(8):847–853, 1995.

[17] M. Anis, S. Areibi, M. Mahmoud, and M. Elmasry "Dynamic and Leakage Power Reduction in MTCMOS Circuits Using an Automated Efficient Gate Clustering," In Proc. DAC 2002.

[18] Liqiong Wei, et al. "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits," In Proc. of the 35th annual conference on Design Automation.

[19] Y. Zhang, et al. "Hotleakage: An architectural, temperature-aware model of subthreshold and gate leakage," Tech. Report CS-2003-05, CS Dept., University of Virginia, Mar. 2003.

[20] J. A. Butts and G. S. Sohi "A static power model for architects." In Proc. of the 33rd MICRO, Dec. 2000.